README

Seizure Simulation Framework

This project provides a simulation pipeline for generating synthetic seizure data using different bifurcation-based dynamotypes, aligned with mechanisms from dynamical systems theory. The simulations aim to replicate neural activity patterns and model the transitions into and out of seizures.

Overview

This repository contains scripts to:

- Simulate seizures based on Hysteresis, Slow Wave, and Piecewise bifurcation dynamics
- Annotate onset and offset times
- Post-process signals with high-pass filtering and additive pink noise
- Output labeled seizure datasets

Cell Output

- *seizures* cell array (before post-processing)
    - Contains the raw simulated seizure data for hysteresis, slow-wave, and piecewise slow-wave and its onset/offset indices.
    - Each entry in post_processed_seizures is a 1×3 cell:
        - Seizure signal
        - Onset index (sample index of seizure start).
        - Offset index (sample index of seizure end).
- *post_processed_seizures* cell array
    - Each entry in post_processed_seizures is a 1×3 cell:
        - Normalized seizure signal with added pink noise (noisy_data_20).
        - Onset index (sample index of seizure start).
        - Offset index (sample index of seizure end).
    - The signal is:
        - Filtered with a high-pass filter (HPF).
        - Normalized between 0 and 1.

    - Adjusted for spike rate
    - Injected with 20% pink noise for realism.

Key Features

- Generates seizures across dynamotype classes
- Uses .fit models to estimate the noise level per bifurcation path point
- Outputs labeled cell arrays: {signal, onset_time, offset_time}
- Allows for parameter sweeping (though only onset/offset curves are currently varied)
- High-pass filtering and pink noise augmentation included in post-processing

Simulation Workflow

1. Choose Generator

You may choose from:

- hysteresis
- Slow wave
- piecewise

Each generator will:

- Select random points on bifurcation onset/offset curves
- Construct a path (arc/circle/piecewise)
- Integrate using the Euler-Maruyama method with stochastic noise
  Save simulated signal with labeled onset/offset times

2. Post-Processing

Run post_processing.m to:

- Normalize signal
- Apply high-pass filtering to model electrode drift

- Add “signal acquisition” noise
- Output: {signal, onset, offset} array of post_processed_seizures

Hyperparameters & Known Challenges

This simulation framework allows users to model seizures via bifurcation dynamics. The core simulation uses fixed hyperparameters (k, k_fast, alpha, dstar, and sigma) and sweeps only over onset and offset curve points by default. This section documents all hyperparameters and the implications of modifying or sweeping them.

Hyperparameters Used

Parameter	Description
k	Controls the speed of the slow variable. Lower k leads to longer seizures. Requires longer tmax.
k_fast	Controls fast subsystem frequency.
alpha	Scales the amplitude of the seizure in the fast subsystem
dstar	Excitibility parameter modulating distance from the resting state
sigma	Noise level injected into the system. Set via .fit files tied to onset/offset bifurcation path points.
tmax	Total simulation time. Must be increased for longer seizures (e.g., with small k).
tstep	Integration time step.
N	Hysteresis branch mode selector. Affects loop behavior (e.g., upper vs. lower branch dynamics).
x0	Initial condition [0; 0; 0]. Start state of the system.

Noise Handling (sigma)

- Noise is dynamically set from .fit models per onset/offset curve point.
- You must modify the code if you want to:
  - Manually specify sigma
  - Use a custom noise distribution

- Key note: Noise is proportional to seizure length and path length. Excessive noise may:
  - Disrupt waveform clarity
  - Cause stuttering in both the main signal and x3
  - Hinder onset/offset detection (especially for hysteresis models in the bistable region)

Sweeping Other Parameters: Known problems

Parameter	Known Challenges
k	Small values → longer seizures → may not finish within tmax. You must increase tmax when sweeping low k.
alpha, k_fast, dstar	Arbitrary values are not always valid. Poor combinations may:

- Fail to generate seizures
- Result in unstable or uninformative dynamics which requires manual tuning and verification

Bistability & Detection Issues

- Hysteresis models involve bistability (rest + seizure attractors). This creates:
  - Path-dependent responses
  - Hysteresis loops
- Under high noise:
  - System may "stutter" near transition thresholds

- Difficult to determine seizure boundaries
- x3 may oscillate near saddle-node bifurcations

File Descriptions

curves.mat

Contains select portions of bifurcation curves for **hysteresis bursts.**

- Each curve is stored as a MATLAB array.
- Example: *SHB* is a *3x62 double* where:
  - **1st dimension**: mu_2
  - **2nd dimension**: mu_1
  - **3rd dimension**: *nu*
- Used in example **Class SN/SH**

curves2.mat

Contains select portions of bifurcation curves for **slow wave bursts.**

- Each curve is stored as a MATLAB array.
- Example: *SNIC* is a *3x44 double* where:
  - **1st dimension**: mu_2
  - **2nd dimension**: mu_1
  - **3rd dimension**: *nu*
- Used in example **Class SNIC/SH**

curves2.mat

Contains **all bifurcation curves** (more comprehensive than `curves.mat` and `curves2.mat`).

- Used for **piecewise** slow wave bursts

test_mesh.mat

- Contains a mesh (vertices and faces) for plotting:
  - Bistable regions
  - Seizure regions
  - Active rest regions

Sphere_mesh.mat

- Contains a mesh (vertices and faces) for plotting the **sphere**.

.fit Files (Hysteresis & Slow Wave Classes)

The effect of **dynamical noise** depends on the path through parameter space because:

1. Integration methods scale differently for different paths.
2. Noise produces varying effects in different dynamical regimes.

To account for this, we empirically modeled noise amplitude for each dynamotype:

- Manually adjusted noise for a subset of paths to achieve 5**-10% deviation in seizure length** (across 10 simulations).
- Fitted a surface (*linearinterp* in MATLAB) to tested points.
- These surfaces assign an appropriate noise amplitude to every simulated path.

This noise model serves to create a relationship between appropriate noise and paths for each class.